

In the Claims

1-2. (Cancelled)

3. (Currently amended) The system of claim [[1]] 7, wherein an unambiguous specification of any descriptor is made through a corresponding descriptor specifier.

4. (Cancelled)

5. (Currently amended) The system of claim [[4]] 7, wherein the second list has a beginning and an end; and

the information is placed at the end of the second list.

6. (Cancelled)

7. (Currently amended) A The system of claim 6 comprising:

a serial bus to connect devices;

the serial bus connected to a first device and a second device; and

a data structure within the first device, the data structure comprising a hierarchy of descriptors, each descriptor identifiable by a descriptor specifier consisting of a list identifier and an object identifier, wherein an extended_information field in a second list contains information for a first list and the second device accesses the extended_information field to move backwards in the hierarchy of descriptors using a command containing the descriptor specifier for one of a first list descriptor and a second list descriptor.

8. (Cancelled)

9. (Currently Amended) The method of claim [[8]] 14, further comprising:

opening the data structure by the first device; and

reading at least one entry in the data structure by [[a]] the second device.

10. (Currently Amended) The method of claim ~~[[8]]~~ 14, comprising:

embedding information about a parent entry within a child list descriptor.

12. (Previously Presented) A method of embedding information about a root_list_ID within a root list descriptor comprising:

reading an extended information field in the root list descriptor by a controller, wherein accessing the extended_information field allows the controller to move backwards in a data structure hierarchy.

13. (Cancelled)

14. (Currently Amended) ~~A The method of claim 13, further comprising:~~

coupling a first device and a second device to a serial bus;

placing into a data structure a descriptor specifier which specifies an entry, the descriptor specifier consisting of a list identifier and an object identifier and the data structure comprising a hierarchy of descriptors as entries, wherein the descriptor specifier is placed in an extended_information field; and

reading the information from the extended_information field to move backwards in the hierarchy.

15-22. (Cancelled)

23. (Currently Amended) ~~A The method of claim 22, further comprising:~~

using a descriptor specifier consisting of a list ID and an object ID to access an entry in a descriptor hierarchy;

placing a parent descriptor info block in an extended_information field; and

accessing the extended_information field to move backwards in the descriptor hierarchy.

24. (Currently Amended) The method of claim ~~21~~ 23, further comprising:

embedding information about a parent entry within a list descriptor.

25. (Currently Amended) The method of claim ~~24~~ 23, further comprising:
placing information about a parent entry in a child list.

26-27. (Cancelled)

28. (Currently Amended) The method of claim ~~27~~ 31, wherein the root list descriptor has
a first position and a second position; and
the descriptor specifier is placed at the second position.

29. (Currently Amended) The method of claim ~~27~~ 31, wherein the child list descriptor has
a third position and a fourth position; and
the descriptor specifier is placed at the fourth position.

30. (Cancelled)

31. (Currently Amended) A ~~The method of claim 30, further~~ comprising:
embedding a parent descriptor info block within a list descriptor, the parent
descriptor info block comprising a descriptor specifier consisting of a list identifier and
an object identifier;
placing the descriptor specifier for the parent descriptor in one of a root list
descriptor and a child list descriptor;
using the descriptor specifier in a descriptor command for opening the parent
entry; and
navigating descriptors in a descriptor hierarchy using the descriptor specifiers,
wherein navigating is in a backward direction.

32. (Previously Presented) A method comprising:
using a delete descriptor command to delete a child list descriptor in a descriptor
hierarchy, the delete descriptor command configured to also update data in a

corresponding parent descriptor and in a corresponding parent list descriptor and delete one of a root list, a child list, and an entry when the child list descriptor is deleted.

33. (Previously Presented) The method of claim 32, further comprising:
deleting a child_ID in the descriptor hierarchy.

34. (Previously Presented) The method of claim 33, further comprising:
updating has_child_ID attributes.

35. (Previously Presented) The method of claim 34, further comprising:
updating entry_descriptor_length.

36. (Previously Presented) The method of claim 35, further comprising:
updating list_descriptor_length.

37. (Currently Amended) The method of claim 32, further comprising:
deleting [[a]] the child list descriptor in the descriptor hierarchy.

38. (Previously Presented) The method of claim 37, further comprising:
deleting the corresponding parent descriptor.

39. (Previously Presented) The method of claim 38, further comprising:
updating no_of__entry descriptors.

40. (Previously Presented) The method of claim 39, further comprising:
updating a list_descriptor_length.

41. (Previously Presented) The method of claim 40, further comprising:
deleting a first child list descriptor in the descriptor hierarchy.

42. (Previously Presented) The method of claim 41, further comprising:
deleting a second child list descriptor.

43. (Previously Presented) The method of claim 42, further comprising:
deleting a list descriptor.

44. (Previously Presented) A system comprising:
a serial bus to connect devices;
the serial bus connected to a first device and a second device; and
a data structure within the first device, the data structure comprising a hierarchy of
descriptors containing a child list descriptor, wherein the second device uses a delete
descriptor command to delete the child list descriptor, the delete descriptor command
configured to also delete the child_ID and one of a root list, a child list, and an entry, and
update one of a has_child_ID attribute, an entry_descriptor_length, and the
list_descriptor_length when the child list descriptor is deleted.